**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**

ЛАБОРАТОРНА РОБОТА № 4

«Flask»

**Виконав студент:**

групи БСД – 22

Муха Максим

Володимирович

**Київ – 2022**

1. Створюємо папку **flask:**

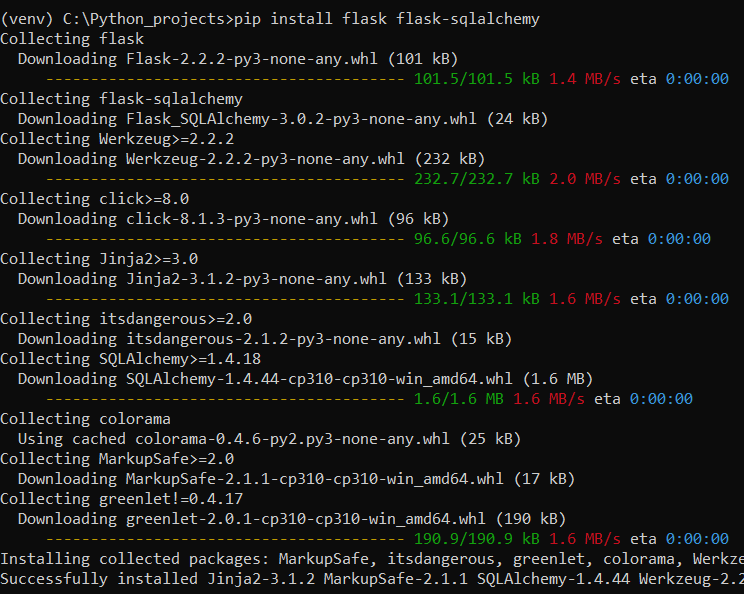


1. Далі, за допомогою команд **pip install virtualenv** встановлюємо віртуальне середовище, а за допомогою команди **virtualenv venv** створюємо папку **venv** де будуть зберігатися усі файли проекту.
2. Активуємо віртуальне середовище командою **.\\flask\venv\Scripts\activate** і також можемо перевірити знаходження папки командою **where**

****

**Рис.1.** Розташування папки

1. Встановлюємо мікрофреймворк **flask** та базу даних **командою pip install flask flask-sqlalchemy**

****

**Рис.2.** Встановлення **flask** та базу даних

1. **Код програми**

from flask import Flask

app = Flask(\_\_name\_\_)

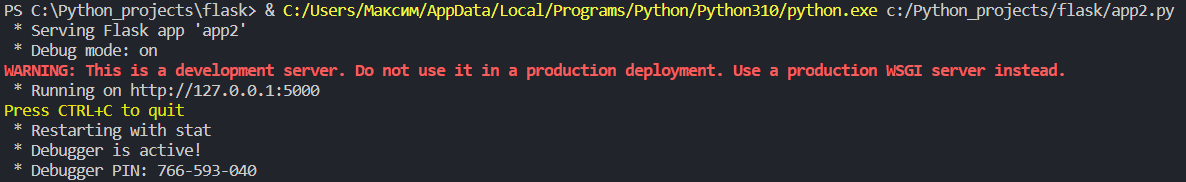
@app.route('/')

def index():

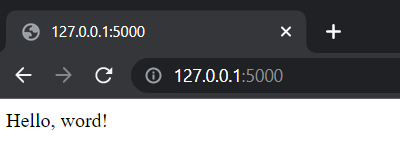
   return "Hello, word!"

if \_\_name\_\_ == "\_\_main\_\_":

   app.run(debug=True)

**1.1**

**Рис.3.** Результат роботи

**1.2**

**Рис.4.** Результат роботи

6) Створюємо папки **static** і **templates.** У папку **templates** додаємо документ **index.html.** Далі трішки редагуємо код, а саме вписуємо рядок

return render\_template('index.html')

Далі в **index.html** записуємо такий код:

<!DOCTYPE *html*>

<html *lang*="en">

<head>

    <meta *charset*="UTF-8">

    <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge">

    <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <h1>Hello world again !</h1>

</body>

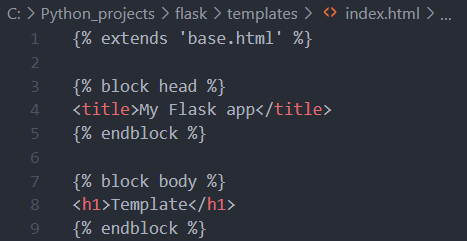
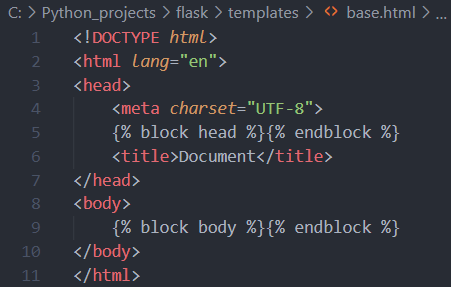
</html>

І дивимось на результат



**Рис.5.** Результат роботи

1. Створюємо файл **base.html**. В цьому шаблоні ми будемо використовувати елементи написані на **Jinja2**, спеціальній мові для шаблонів у **Flask**

1. Імпортуємо модуль **SQLALCHEMY** і програма буде виглядати вже так:

from flask import Flask, render\_template, url\_for

from flask\_sqlalchemy import SQLAlchemy

from datetime import datetime

app = Flask(\_\_name\_\_)

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///test.db'

db = SQLAlchemy(app)

class Todo(db.Model):

    id = db.Column(db.Integer, primary\_key=True)

    content = db.Column(db.String(200), nullable=False)

    data\_created = db.Column(db.DateTime, default=datetime.utcnow)

    def \_\_repr\_\_(self) -> str:

       return '<Task %r>' % self.id

@app.route('/')

def index():

   return render\_template('index.html')

if \_\_name\_\_ == "\_\_main\_\_":

   app.run(debug=True)

1. Редагуємо вміст файлу **index.html**

{% extends 'base.html' %}

{% block head %}

<title>Task Manager</title>

<link *rel*="stylesheet" *href*="{{ url\_for('static', filename='css/main.css') }}">

{% endblock %}

{% block body %}

<div *class*="content">

   <h1>Task Manager</h1>

   <table>

    <tr>

        <th>Task</th>

        <th>Added</th>

        <th>Actions</th>

    </tr>

    {% for task in tasks %}

        <tr>

            <td>{{ task.content }}</td>

            <td>{{ task.data\_created.date() }}</td>

            <td>

                <a *href*="">Delete</a>

                <br>

                <a *href*="">Update</a>

            </td>

        </tr>

    {% endfor %}

   </table>

   <form *action*="/" *method*="POST">

    <input *type*="text" *name*="content" *id*="content">

    <input *type*="submit" *value*="Add Task">

    </form>

</div>

{% endblock %}

Також редагуємо файл **app2.py**

import datetime

from flask import Flask, render\_template, request, redirect

from flask\_sqlalchemy import SQLAlchemy

app = Flask(\_\_name\_\_)

app.config['SQLALCHEMY\_DATABASE\_URI'] = 'sqlite:///test.db'

db = SQLAlchemy(app)

app.app\_context().push()

class Todo(db.Model):

   id = db.Column(db.Integer, primary\_key=True)

   content = db.Column(db.String(200), nullable=False)

   data\_created = db.Column(db.DateTime, default=datetime.utcnow)

   def \_\_repr\_\_(self) -> str:

      return '<Task %r>' % self.id

@app.route('/', methods=['POST', 'GET'])

def index():

   if request.method == 'POST':

       task\_content = request.form['content']

       new\_task = Todo(content=task\_content)

       try:

           db.session.add(new\_task)

           db.session.commit()

           return \_RedirectStream('/')

       except:

           return 'There was an issue adding your task'

   else:

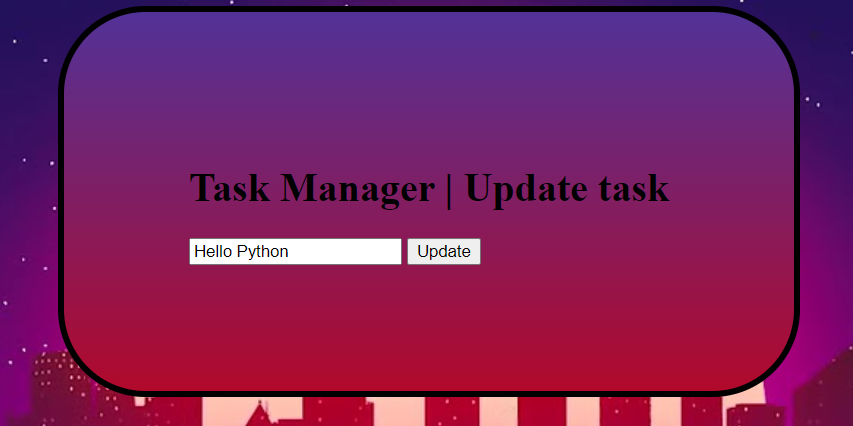
       tasks = Todo.query.order\_by(Todo.data\_created).all()

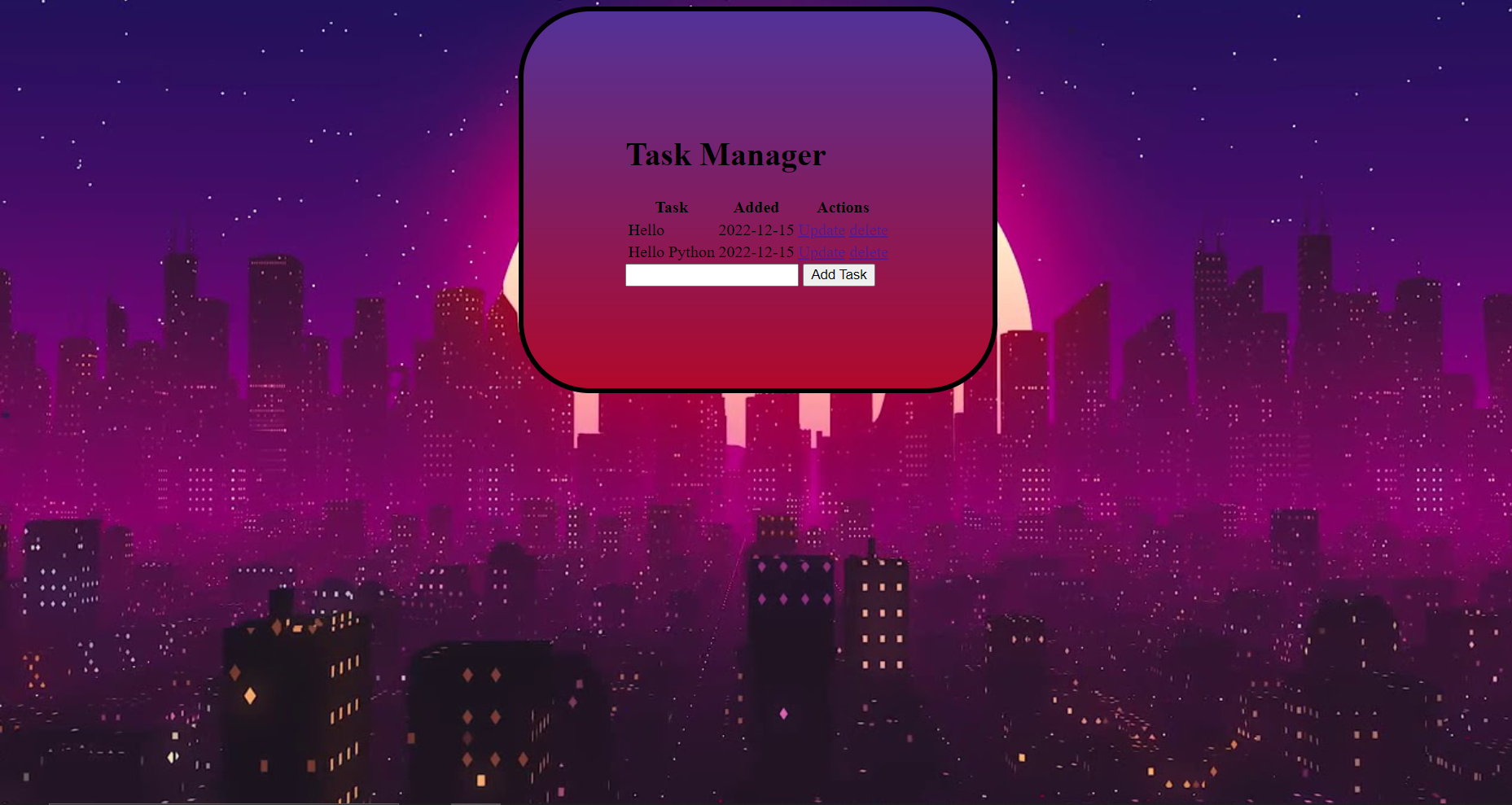
       return render\_template('index.html', tasks=tasks)

if \_\_name\_\_ == "\_\_main\_\_":

   app.run(debug=True)

**Результат роботи**

****



**Висновок** В даній лабораторній роботі, я отримав навички в створенні веб-ресурсів.